

# Apuntes de Algoritmos Genéticos

## Parte III: Cruzamiento

M. Angélica Pinninghoff J. - Ricardo Contreras A.  
Magíster en Inteligencia Artificial  
Universidad Adolfo Ibáñez

2022

### 1 Preliminares

El Cruzamiento es uno de los operadores fundamentales en los Algoritmos Genéticos. Durante el proceso de selección se elige la cantidad de padres de acuerdo al tamaño de la población. Una vez que ha finalizado el proceso de selección, se forman las parejas que van a reproducirse, generalmente las parejas se forman por contiguidad. De cada pareja se generan dos hijos, por lo tanto el número de hijos generados es igual al número de padres seleccionados. Una vez que están formadas las parejas de padres se ejecuta el cruzamiento que consiste en el intercambio de material genético entre dos cromosomas (los padres) para obtener como resultado otros dos nuevos cromosomas (los hijos).

La figura 1 esquematiza estos conceptos.

El cruzamiento de una pareja se realiza con una cierta probabilidad, llamada probabilidad de cruzamiento ( $p_c$ ). Esta probabilidad es un parámetro en los algoritmos genéticos y normalmente es alta, sobre el 60%. Por ejemplo, supóngase un  $p_c = 80\%$ ; por cada pareja se genera un número aleatorio,  $r$ , entre 1 y 100. Si  $r \leq 80$  entonces la pareja se cruza y genera dos hijos. Si  $r > 80$  entonces la pareja no se cruza y ambos padres pasan, sin modificar, a la generación siguiente. Existen muchos métodos u operadores de cruzamiento. A continuación se verán los más comunes.

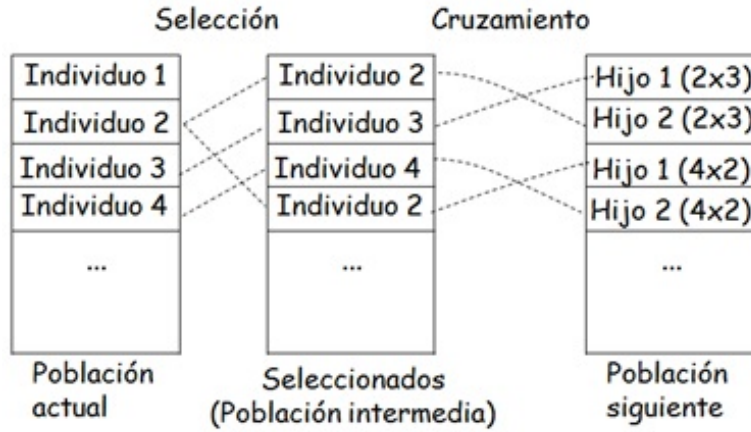


Figure 1: Selección y formación de parejas.

## 2 Cruzamiento en representación binaria

En la representación binaria los genes sólo contienen ceros o unos. Por ejemplo, en el problema de la mochila, que consiste en maximizar el número de ítems que caben en una mochila de cierta capacidad, a cada ítem se le puede asignar una posición dentro del cromosoma (un gen) y el contenido de ese gen puede ser 0 si el ítem no está en la mochila o 1 si está.

**Cruzamiento en 1-punto:** Se tienen 2 padres que llamaremos Padre1 y Padre2. Se elige, para esta pareja, un número aleatorio, entre 1 y el tamaño del cromosoma. Este número, es el punto de cruce. Los hijos, Hijo1 e Hijo2, se forman con la misma información de sus respectivos padres desde el comienzo hasta el punto de cruce; luego intercambian el resto, desde el punto de cruce hasta el final (ver ejemplo en figura 2).

**Cruzamiento en 2-puntos:** Es similar al anterior, pero se tiene 2 puntos de cruce y los hijos intercambian la información de los padres sólo entre estos dos puntos de cruce (ver ejemplo en figura 3). Este tipo de cruzamiento puede ser generalizado a n-puntos, siguiendo un procedimiento similar.

**Cruzamiento uniforme:** Para formar los hijos, se recorre todo el cromosoma de los padres decidiendo al azar si los contenidos en cada gen son intercambiados o no (ver ejemplo en figura 4).

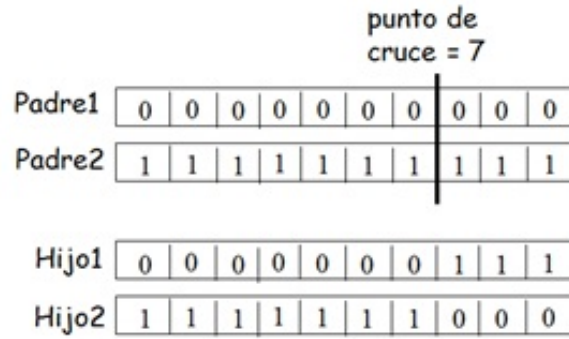


Figure 2: Cruzamiento en un punto.

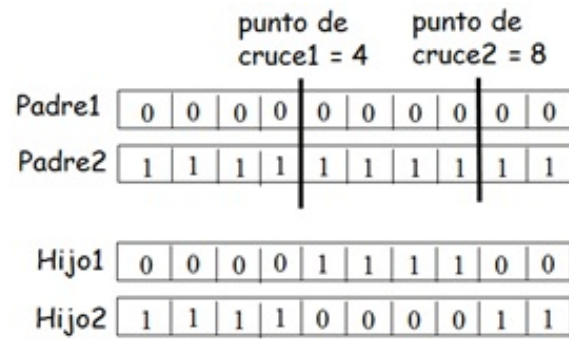


Figure 3: Cruzamiento en dos puntos.

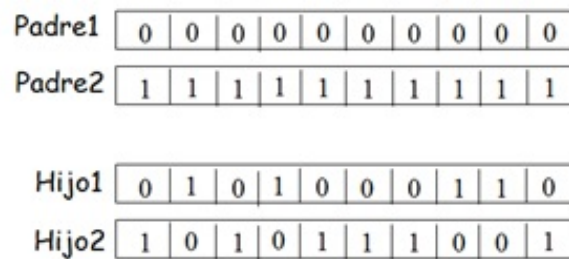


Figure 4: Cruzamiento uniforme.

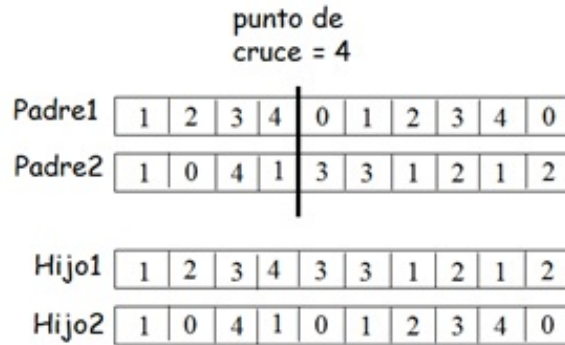


Figure 5: Cruzamiento en un punto en representación entera.

Los operadores de cruzamiento anteriores también pueden usarse en otro tipo de representaciones como representaciones numéricas (reales o enteros), donde los valores de los genes pueden estar repetidos. Por ejemplo en coloración de mapas o coloración de grafos, donde cada color puede ser representado por un número entero y los lugares geográficos del mapa o los nodos de un grafo, pueden estar representados por las posiciones de los genes dentro del cromosoma. En la figura 5 se muestra un ejemplo de cruzamiento en 1-punto en cromosomas con enteros.

Se puede observar que el número 0 desaparece en el Hijo1, producto del cruzamiento.

### 3 Cruzamiento en ordenamiento

En algunos problemas, al aplicar alguno de los operadores anteriores, el cruzamiento da lugar a cromosomas que codifican soluciones no válidas; en estos casos hay que aplicar un operador de cruzamiento de forma que genere siempre soluciones válidas, pero que mantenga la idea básica de mezclar valores de los genes de los padres. Es el caso de los problemas cuya representación son permutaciones u ordenamiento de números. Por ejemplo el problema del vendedor viajero: un vendedor viajero tiene que visitar  $n$  clientes ubicados en distintos puntos geográficos. El vendedor parte desde un punto, recorre todos los clientes, sin repetir, y vuelve a su punto de origen. Hay distintas formas de hacer el recorrido, en realidad hay  $n!$  posibles recorridos. El recorrido óptimo es aquel que tiene el costo (o la distancia) mínimo. A cada cliente se le puede asignar un número entero de  $1$  a  $n$ .

1	2	3	4	5	6	7	8	9	0
1	0	4	8	3	9	5	2	7	6

Figure 6: Ordenamientos.

Como cada cliente debe ser visitado una sola vez, el número que lo representa aparecerá una sola vez en el recorrido; y como el vendedor viajero debe visitar a todos los clientes, en un recorrido deberán aparecer todos los números. Entonces, un cromosoma representa un recorrido del vendedor viajero; la secuencia de números muestra el orden en que el vendedor visita a sus clientes. En la figura 6 se muestran dos recorridos: en el primero se visitaron los clientes en el orden asignado; en el segundo, primero se visitó el cliente 1, luego del cliente 1 el vendedor fue donde el cliente 0, luego donde el cliente 4 y así sucesivamente. Si a estos cromosomas les aplicamos cruzamiento en un punto, por ejemplo con punto de cruce = 2, entonces en el primer hijo estaría repetido el número 2 y no estaría el 0, por lo tanto ya no sería un recorrido válido. El segundo hijo tampoco sería recorrido válido, ya que tendría repetido el 0 y le faltaría el 2.

Existen algunos métodos de cruzamiento que permiten mezclar los cromosomas sin que se pierda su calidad de ordenamiento. Los operadores de este tipo más conocidos son, entre otros, el OX (Order Crossover), PMX (Partially-Mapped Crossover) y CX (Cycle Crossover) [3], de los cuales sólo nos preocuparemos del primero, OX. En el operador de cruzamiento OX, se eligen aleatoriamente dos puntos de cruce, posteriormente el proceso de cruzamiento se realiza en tres pasos:

**Paso1:** Se eligen en forma aleatoria 2 puntos de cruce y se crea la estructura de los hijos, manteniendo los valores de sus respectivos padres entre ambos puntos de cruce, como se muestra en la figura 7.

**Paso2:** Se crea una secuencia por cada padre. Estas contienen los valores de los genes de cada padre, comenzando con los genes desde el segundo punto de cruce hasta el final del cromosoma y continuando con el comienzo del mismo cromosoma hasta el segundo punto de cruce. Una vez obtenidas las secuencias de esta forma, en la secuencia  $S2'$  obtenida del *Padre2* se eliminan los valores que aparecen en el  $Hijo1'$ , generando la secuencia  $S2$ . De la misma manera, en la secuencia  $S1'$  obtenida del *Padre1* se eliminan los valores que aparecen en el  $Hijo2'$ , generando la secuencia  $S1$ . Este proceso se muestra en la figura 8.

	pcruce1= 3			pcruce2 = 6						
Padre1	1	2	3	4	5	6	7	8	9	0
Padre2	1	0	4	8	3	9	5	2	7	6
Hijo1'				4	5	6				
Hijo2'				8	3	9				

Figure 7: Paso 1, en cruzamiento OX.

S1'	7	8	9	0	1	2	3	4	5	6
S2'	5	2	7	6	1	0	4	8	3	9
S1	7	0	1	2	4	5	6			
S2	2	7	1	0	8	3	9			

Figure 8: Paso 2, secuencia de los padres generadas a partir del segundo punto de cruce.

Padre1	1	2	3	4	5	6	7	8	9	0
Padre2	1	0	4	8	3	9	5	2	7	6
Hijo1	8	3	9	4	5	6	2	7	1	0
Hijo2	4	5	6	8	3	9	7	0	1	2

Figure 9: Paso 3, resultado después de aplicar el operador OX.

En  $S1'$  se eliminaron el 8, 2 y 9. En  $S2'$  se eliminaron el 4, 5 y 6.

**Paso3:** Finalmente, se rellenan los hijos, a partir del segundo punto de cruce, con la secuencia del padre contrario. En la figura 9 se muestran los padres con sus hijos resultantes. Como se puede ver, los hijos son diferentes a los padres pero son ordenamientos válidos.

## References

- [1] Goldberg, David E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley Longman Publishing Co.
- [2] Holland, John (1975). Adaptation in Natural and Artificial Systems. The MIT Press.
- [3] Michalewicz, Z., Fogel, D. How to Solve It: Modern Heuristics. Springer, 2000.
- [4] Mitchell, Melanie (1998). An Introduction to Genetic Algorithms. Cambridge, Mass: MIT Press.
- [5] Parmee, Ian C. (2001). Evolutionary and Adaptative Computing in Engineering Design. Springer.